



**PATENT APPLICATION**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re application of

Docket No: A8519

William J. BAER, et al.

Application No.: 09/488,971

Group Art Unit: 2176

Confirmation No.: 5172

Examiner: William L. BASHORE

Filed: January 21, 2000

For: **METHOD AND SYSTEM FOR MOVING CONTENT IN A CONTENT OBJECT  
STORED IN A DATA REPOSITORY**

**SUBMISSION OF APPEAL BRIEF**

**MAIL STOP APPEAL BRIEF - PATENTS**

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

Sir:

Submitted herewith please find an Appeal Brief. The USPTO is directed and authorized to charge the statutory fee of \$500.00 and/or all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account. A duplicate copy of this paper is attached.

Respectfully submitted,

10/18/2005 JADDO1 00000019 194880 09488971

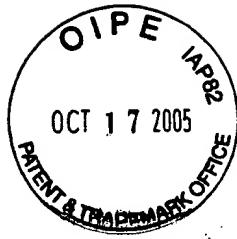
01 FC:1402 500.00 DA

SUGHRUE MION, PLLC  
Telephone: (202) 293-7060  
Facsimile: (202) 293-7860

WASHINGTON OFFICE  
**23373**  
CUSTOMER NUMBER

  
Ruthleen E. Uy  
Registration No. 51,361

Date: October 17, 2005



**PATENT APPLICATION**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re application of

Docket No: A8519

William J. BAER, et al.

Application No.: 09/488,971

Group Art Unit: 2176

Confirmation No.: 5172

Examiner: William L. BASHORE

Filed: January 21, 2000

For: **METHOD AND SYSTEM FOR MOVING CONTENT IN A CONTENT OBJECT  
STORED IN A DATA REPOSITORY**

**APPEAL BRIEF UNDER 37 C.F.R. § 41.37**

**MAIL STOP APPEAL BRIEF - PATENTS**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 41.37, Appellant submits the following:

**Table of Contents**

I.	REAL PARTY IN INTEREST.....	2
II.	RELATED APPEALS AND INTERFERENCES.....	3
III.	STATUS OF CLAIMS .....	4
IV.	STATUS OF AMENDMENTS .....	5
V.	SUMMARY OF THE CLAIMED SUBJECT MATTER .....	6
VI.	GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL .....	12
VII.	ARGUMENT.....	13
VIII.	CONCLUSION.....	20
	CLAIMS APPENDIX.....	21
	EVIDENCE APPENDIX:.....	30
	RELATED PROCEEDINGS APPENDIX .....	31

**I. REAL PARTY IN INTEREST**

The real party in interest in this appeal is International Business Machines Corporation (“IBM”) of Armonk, New York, the assignee.

**II. RELATED APPEALS AND INTERFERENCES**

The following applications are the subject of a prior or pending appeal, are related to the present application on appeal, and may directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

- U.S. Application No. 09/488,969: Notice of Appeal filed April 28, 2005; Appeal Brief filed June 28, 2005.
- U.S. Application No. 09/489,561: Notice of Appeal filed January 18, 2005; finality of the rejections was withdrawn and a Notice of Allowance mailed on April 27, 2005.
- U.S. Application No. 09/489,605: Notice of Appeal filed January 22, 2003; Appeal Brief filed March 24, 2003; and a Notice of Allowance mailed on April 27, 2005.
- U.S. Application No. 09/489,143: Notice of Appeal filed June 2, 2005; Appeal Brief filed August 2, 2005.

There are no other appeals or interferences known to Appellant, Appellant's legal representative, or the assignee that will directly affect or be directly affected by, or have a bearing on, the Board's decision in this appeal.

**III. STATUS OF CLAIMS**

Claims 1-43 are pending in the present application and stand rejected.

Claims 1-27 and 40-43 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over ezWriter 2.0 for Windows, August 5, 1998 by Lance Vaughn, Atlanta, IN, downloaded from <http://www.winsite.com> on 6/22/2003, application screenshots pages 1-15 (hereinafter “eZWriter”), in view of Bromberg et al. (U.S. Patent No. 6,529,889; hereinafter “Bromberg”).

Claims 28-39 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over ezWriter and Bromberg and further in view of Poole et al. (U.S. Patent No. 6,006,242; hereinafter “Poole”).

A copy of the claims on appeal is set forth in an attached Appendix.

**IV. STATUS OF AMENDMENTS**

Amendments to the claims were submitted in an Amendment Under 37 C.F.R. § 1.114(c) filed June 25, 2004. All amendments are believed to have been previously entered and made of record. A Response under 37 C.F.R. § 1.111 was filed on December 22, 2004, in response to the Non-Final Office Action dated September 22, 2004. A Response under 37 C.F.R. § 1.116 was filed on July 18, 2005, in response to the Final Office Action dated May 18, 2005. In an Advisory Action dated August 11, 2005, the Examiner states that the Response filed July 18, 2005, has been considered but did not place the application in a condition for allowance.

**V. SUMMARY OF THE CLAIMED SUBJECT MATTER**

Appellant's invention as recited in, for example, independent claims 1, 4, 8, 11, 15, 18, 40 and 42, is related to methods, program storage devices, and systems for reordering content in a content object.

Content management systems such as relational databases, digital libraries, and media servers have enabled contents of all types to be stored digitally. Content management systems also have functions for manipulating the stored content. An exemplary embodiment of the present invention permits a user to take advantage of vast stores of content to create compilations tailored to the user's needs or desires. See page 1 of Appellant's specification.

**Claim 1**

A method for reordering content in a content object (e.g. custom compilation, custom book outline) stored as a plurality of hierarchically related content entities in a data repository (see e.g. Fig. 1, digital library 20; page 19, lines 8-9), each content entity having an identifier (e.g. title), comprising the steps of defining the content object with a list of content entity identifiers such that moving a content entity identifier to a new location (see page 5, lines 3-5; see Fig. 16, element 168) within the list redefines the order of the object's content entities (see page 4, lines 24-30; page 93 lines 1-25), wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities (page 4, lines 1-5; page 21, line 27 to page 22, line 5).

**Claim 4**

A method for reordering content in a hierarchically structured content object (e.g. custom compilation, custom book outline) stored as a plurality of hierarchically related content entities in a data repository (see e.g. Fig. 1, digital library 20; page 19, lines 8-9), each content entity having an identifier (e.g. title), comprising the steps of defining the content object with an outline of containers and content entity identifiers such that moving a container or content entity identifier to a new location (see page 5, lines 3-5; see Fig. 16, element 168) within the outline redefines the order of the object's content entities (see page 4, lines 24-30; page 93 lines 1-25), wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities (page 4, lines 1-5; page 21, line 27 to page 22, line 5).

**Claim 8**

A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for reordering content in a content object (e.g. custom compilation, custom book outline) stored as a plurality of hierarchically related content entities in a data repository (see e.g. Fig. 1, digital library 20; page 19, lines 8-9), each content entity having an identifier (e.g. title), comprising the steps of defining the content object with a list of content entity identifiers such that moving a content entity identifier to a new location (see page 5, lines 3-5; see Fig. 16, element 168) within the list redefines the order of the object's content entities (see page 4, lines 24-30; page 93 lines 1-25), wherein the hierarchically related content entities further comprise a parent container type and a

child container type, wherein parent containers can contain child containers, and child containers can contain content entities (page 4, lines 1-5; page 21, line 27 to page 22, line 5).

**Claim 11**

A method for reordering content in a hierarchically structured content object (e.g. custom compilation, custom book outline) stored as a plurality of hierarchically related content entities in a data repository (see e.g. Fig. 1, digital library 20; page 19, lines 8-9), each content entity having an identifier (e.g. title), comprising the steps of defining the content object with an outline of containers and content entity identifiers such that moving a container or content entity identifier to a new location (see page 5, lines 3-5; see Fig. 16, element 168) within the outline redefines the order of the object's content entities (see page 4, lines 24-30; page 93 lines 1-25), wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities (page 4, lines 1-5; page 21, line 27 to page 22, line 5).

**Claim 15**

A system for reordering content in a content object (e.g. custom compilation, custom book outline) stored as a plurality of hierarchically related content entities in a data repository (see e.g. Fig. 1, digital library 20; page 19, lines 8-9), each content entity having an identifier (e.g. title), comprising means for defining the content object with a list of content entity identifiers (e.g. ECBOMoveContent function, ECBORelocateContent function; see specification at page 94, line 3 to page 96, line 21) such that moving a content entity identifier to a new location (see page 5, lines 3-5; see Fig. 16, element 168) within the list redefines the order of the

object's content entities (see page 4, lines 24-30; page 93 lines 1-25), wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities (page 4, lines 1-5; page 21, line 27 to page 22, line 5).

**Claim 18**

A system for reordering content in a hierarchically structured content object (e.g. custom compilation, custom book outline) stored as a plurality of hierarchically related content entities in a data repository (see e.g. Fig. 1, digital library 20; page 19, lines 8-9), each content entity having an identifier (e.g. title), comprising means for defining the content object with an outline of containers and content entity identifiers (e.g. ECBOMoveContent function, ECBORelocateContent function; see specification at page 94, line 3 to page 96, line 21) such that moving a container or content entity identifier to a new location (see page 5, lines 3-5; see Fig. 16, element 168) within the outline redefines the order of the object's content entities (see page 4, lines 24-30; page 93 lines 1-25), wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities (page 4, lines 1-5; page 21, line 27 to page 22, line 5).

**Claim 40**

A method for reordering content in a content object (e.g. custom compilation, custom book outline) stored as a plurality of hierarchically related content entities in a data repository (see e.g. Fig. 1, digital library 20; page 19, lines 8-9), each content entity having an identifier

(e.g. title), comprising defining the content object with a first list of content entity identifiers such that moving a content entity identifier from the first list of content entity identifiers to a new location (see page 5, lines 3-5; see Fig. 16, element 168) within the first list of content entity identifiers redefines the order of the object's content entities (see page 4, lines 24-30; page 93 lines 1-25), wherein moving a content entity identifier in the first list (e.g. My Book Outline) to a new location comprises selecting the content entity identifier from the first list of content entity identifiers to be moved; and specifying a location from a second list of content entity identifiers where the content entity identifier from the first list of content entity identifiers is to be moved; wherein the specified location comprises at least one of a current content entity identifier or a newly created content entity identifier (page 4, lines 1-5; page 21, line 27 to page 22, line 5).

**Claim 42**

A method for reordering content in a customized electronic book (e.g. custom compilation, custom book outline) stored as a plurality of hierarchically related content entities in a computer database (see e.g. Fig. 1, digital library 20; page 19, lines 8-9), each content entity having an identifier (e.g. title), comprising defining the customized book with a first outline of content entity identifiers (e.g. My Book Outline) such that moving a content entity identifier from the first outline to a new location (see page 5, lines 3-5; see Fig. 16, element 168) within the first outline of content entity identifiers redefines the order of the content entities within the customized book (see page 4, lines 24-30; page 93 lines 1-25), wherein moving a content entity identifier from the first outline to a new location comprises selecting a content entity identifier from the first outline to be moved; and specifying a location from a second outline of

content entity identifiers where the content entity identifier of the first outline is to be moved; wherein the specified location comprises at least one of a current content entity identifier or a newly created content entity identifier (See Fig. 10A; page 4, lines 1-5; page 21, line 27 to page 22, line 5).

**VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

1. Claims 1-27 and 40-43 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over ezWriter in view of Bromberg.
2. Claims 28-39 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over ezWriter and Bromberg and further in view of Poole.

**VII. ARGUMENT**

**1. Claims 1-27 and 40-43 are patentable over ezWriter in view of Bromberg**

As noted above, claims 1-27 and 40-43 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over ezWriter in view of Bromberg. It is respectfully submitted that claims 1-27 and 40-43 are patentable over ezWriter in view of Bromberg for at least the following reasons.

**Claim 1**

Claim 1 recites “a method for reordering content in a content object stored as a plurality of hierarchically related content entities in a data repository, each content entity having an identifier.” The method includes “defining the content object with a list of content entity identifiers such that moving a content entity identifier to a new location within the list redefines the order of the object’s content entities.” ezWriter is cited for teaching these aspects of claim 1.

Claim 1 also recites “wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities.” The Examiner acknowledges that ezWriter does not teach this aspect of claim 1 and cites Bromberg to cure the deficiency.

ezWriter is a utility program “to assist with the organization of outlines, concepts, notes, chapters, etc.” See ezWriter p. 2 It uses two types of files, a series of rich text format (RTF) files that hold data entered by a writer, such as notes and other text (*Id.*) and an ezWriter file (.ewz) which is an ASCII text file that “is just an index of (or series of pointers to)” the RTF

files. See ezWriter p. 7. In describing “[w]hat’s so cool about ezWriter,” the reference states that there will be “[n]o more getting caught up in the cumbersome environment of full-blown word-processors -- ezWriter is a streamlined environment developed specifically for writers!” See ezWriter p. 3. The ezWriter file (.ezw) contains an outline of entities corresponding to the RTF files and uses periods preceding each entry to indicate the relationship between the various entries, as stated at page 7 and reproduced below.

As you can see, the periods are used to indicate the relationship between the various entries. Any entry with only one preceding period resides at the top level. All entries must have at least one period. To create a child entry of another entry, just add it beneath the particular entry with one extra period.

.A Parent Entry  
..A Child of That Entry  
..Another Child of That Entry

The Examiner asserts that Bromberg, in showing a topic hierarchy and a display for displaying questions related to the topic, discloses hierarchically related content entities comprising a parent container type that can contain a child container type. Even assuming for the moment that Bromberg discloses such parent and child container types, the topics (or “content entities”) of Bromberg are different than the ezWriter .rtf files which are asserted to be the claimed “content entities.” Consequently, it is respectfully submitted that, as would be apparent to one of ordinary skill in the art, it would not have been obvious at the time of the invention to modify the ezWriter to use the parent and child container types instead of the .rtf files disclosed in ezWriter.

Also, modifying ezWriter to use Bromberg's parent and child containers in place of the .rtf files disclosed in ezWriter would destroy the principle operation of ezWriter, and hence, would not have been an obvious modification to a person of ordinary skill at the time of the invention. See MPEP 2143.01 "The Proposed Modification Cannot Change the Principle of Operation of a Reference." ezWriter, in keeping with its stated goal of providing a "streamlined environment developed especially for writers," uses only a sequence of periods preceding the entries in the ezWriter file to indicate the relationships between those entries such that "to create a child entry of another entry, just add it beneath the particular entry with one extra period." See ezWriter p. 8.

It is respectfully submitted that ezWriter does not teach or even suggest the use of containers, much less that a parent container can contain a child container, as required by claim 1 for example. To modify ezWriter to abandon its use of the sequence of periods in front of the entries in the .ezw file and instead use parent and child containers in which a parent container can contain a child container, would fundamentally change the principle of operation of how ezWriter represents relationships among entities. Accordingly, it would not have been obvious to modify ezWriter to replace its technique of using a sequence of periods to represent relationships with the parent/child containers of Bromberg.

The Examiner's asserted motivation for modifying ezWriter to use the parent/child containers of Bromberg is that the rollup feature disclosed in Bromberg could be used to make "set wide" changes throughout the ezWriter set of files. It is respectfully submitted, however, that the prior art neither teaches nor suggests such a motivation. The Examiner asserts that

Bromberg discloses that the rollup feature that accompanies the use of parent/child containers can be used to allow a parent container to contain information on the activity of containers that are subordinate to it (child containers). However, there is no suggestion of making set wide changes nor does the prior art suggest a need to make such changes.

In addition, there is no teaching or suggestion in ezWriter for such a modification, particularly since ezWriter automatically saves documents after edits. Modifying ezWriter to contain the rollup feature of Bromberg would clearly result in a substantial modification of the principle of operation of ezWriter, and would introduce a level of complexity that is contradictory to its goal of providing “a streamlined environment.” See ezWriter under the heading “What’s so cool about ezWriter?”. This evidences that the Examiner’s reasoning is merely a result of impermissible hindsight. Accordingly, it is respectfully submitted that there is no teaching or suggestion in the prior art to make the asserted combination.

Further, there is no reason why the child containers of Bromberg, which contain questions to be answered by an expert, would be combined with the .rtf files of ezWriter (i.e. “A Parent Entity,” “A Child of that Entity,” and “Another Child of that Entity”).

Bromberg discloses containers of information regarding topics to be stored in a knowledge database and respective questions to be asked of an expert. The questions and answers are stored in the knowledge database according to topic. The Examiner asserts that it would be obvious to combine the parent container (topics) and child containers (questions) of Bromberg with the .rtf word processing files of ezWriter. However, there is no teaching or

suggestion that ezWriter should be modified to include the organization of parent containers (regarding topics of discussion), and child containers containing (questions to be asked regarding a particular topic). Bromberg teaches such an organization so as to cure deficiencies in the prior art where irrelevant questions were asked of a user on particular topics. See Bromberg col. 1, line 57 to col. 2, lines 5. ezWriter does not suffer from this problem.

Although ezWriter (page 8) discloses “A Parent Entity,” “A Child of that Entity,” and “Another Child of that Entity,” this aspect of ezWriter was cited by the Examiner for teaching “a plurality of hierarchically related content entities of a content object.” There is no teaching or suggestion in the prior art that the hierarchically related content entities of ezWriter should use a parent container and a child container. Further, there is no teaching or suggestion in the prior art that a parent container contain a child container or that the child containers can contain content entities (i.e. “A Parent Entity,” “A Child of that Entity,” and “Another Child of that Entity” as cited by the Examiner), as required by the claim.

The Examiner asserts that ezWriter’s index file maps each .rtf file (one file per section) into a hierarchical tree with parent and child relationships, citing page 8 of ezWriter in support. Further, the Examiner states that the combination of the parent and child containers of Bromberg with the parent and child tree representation of ezWriter teaches or suggests the Appellant’s claimed invention. Assuming *arguendo*, that the combination of Bromberg with ezWriter meets all the limitations of the claim, it is respectfully submitted that it would not have been obvious to combine the references as the Examiner asserts. In particular, it would not be obvious to one of

skill in the art to combine the .rtf (rich text format) word processing files of ezWriter with the parent and child container types of Bromberg.

For at least the above reasons, it is respectfully submitted that the combination of ezWriter and Bromberg would not have been obvious. Therefore, claims 1, 4, 8, 11, 15, 18 and their dependent claims should be deemed patentable.

**Claim 40 and 42**

Claim 40 recites that “moving a content entity identifier in the first list to a new location comprises: selecting the content entity identifier from the first list of content entity identifiers to be moved; and specifying a location from a second list of content entity identifiers where the content entity identifier from the first list of content entity identifiers is to be moved; wherein the specified location comprises at least one of a current content entity identifier or a newly created content entity identifier” which is not recited in claim 1. Consequently, the Examiner’s assertion for the rejection of claim 1 are not applicable to all the elements of claim 40.

In particular, there is no teaching or suggestion in the combination of ezWriter and Bromberg of “a second list of content entity identifiers.” ezWriter discloses a single list of content entity identifiers. See for example page 8. The order of the content entities within the list of content entity identifiers is modified by the addition or removal of periods. At no point is the movement of a content entity identifier performed by “specifying a location from *a second list of content entity identifiers* where the content entity identifier from the first list of content entity identifiers is to be moved,” as required by claim 40.

Moreover, there is no teaching or suggestion that a location of a second list of content entity identifiers comprises a current content entity identifier or a newly created content entity identifier as required by claim 40. Further, any suggestion by the Examiner to modify ezWriter to include the elements of claim 41 would clearly be a result of impermissible hindsight.

For at least the above reasons, claim 40 and its dependent claim should be deemed allowable. To the extent claim 42 recites similar subject matter, claim 42 and its dependent claim should be deemed allowable for the same reasons.

**2. Claims 28-39 are patentable over ezWriter and Bromberg and further in view of Poole**

Claims 28-39 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over ezWriter and Bromberg and further in view of Poole. Claims 28-39 should be deemed patentable by virtue of their dependency to independent claims 1, 4, 8, 11, 15, and 18 for at least the reasons set forth above.

Poole is directed to an apparatus and method for dynamically creating a document. In particular, Poole teaches a document production system that produces for a high degree of content re-use such that a base of content can be used to construct documents, such as boilerplate insurance and bank documents. See col. 2, lines 2-7. Consequently, Poole does not satisfy the deficiencies of ezWriter and Bromberg discussed above. Therefore, claims 28-39 should further be deemed patentable.

**VIII. CONCLUSION**

Unless a check is submitted herewith for the fee required under 37 C.F.R. §41.37(a) and 1.17(c), please charge said fee to Deposit Account No. 19-4880.

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,

  
\_\_\_\_\_  
Ruthleen E. Uy  
Registration No. 51,361

SUGHRUE MION, PLLC  
Telephone: (202) 293-7060  
Facsimile: (202) 293-7860

WASHINGTON OFFICE  
**23373**  
CUSTOMER NUMBER

Date: October 17, 2005

**CLAIMS APPENDIX**

**CLAIMS 1-43 ON APPEAL:**

1. A method for reordering content in a content object stored as a plurality of hierarchically related content entities in a data repository, each content entity having an identifier, comprising the steps of:

defining the content object with a list of content entity identifiers such that moving a content entity identifier to a new location within the list redefines the order of the object's content entities,

wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities.

2. The method of claim 1, further comprising the step of receiving a user-specification of a content entity to move and target location for the specified content entity.

3. The method of claim 2, further comprising the step of providing a user interface communicating with the data repository, and providing a mechanism for enabling a user to select a content entity to move and specify the target location in the content object through the user interface.

4. A method for reordering content in a hierarchically structured content object stored as a plurality of hierarchically related content entities in a data repository, each content entity having an identifier, comprising the steps of:

defining the content object with an outline of containers and content entity identifiers such that moving a container or content entity identifier to a new location within the outline redefines the order of the object's content entities,

wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities.

5. The method of claim 4, further comprising the step of receiving a user-specification of a content entity to move and target location for the specified content entity.

6. The method of claim 5, further comprising the step of providing a user interface communicating with the data repository, and providing a mechanism for enabling a user to select a content entity to move and specify the target location in the content object through the user interface.

7. The method of claim 4, wherein the content object comprises a book, the content entities comprise sections and the containers comprise chapters and books.

8. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for reordering content in a content object stored as a plurality of hierarchically related content entities in a data repository, each content entity having an identifier, comprising the steps of:

defining the content object with a list of content entity identifiers such that moving a content entity identifier to a new location within the list redefines the order of the object's content entities,

wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities.

9. The method of claim 8, further comprising the step of receiving a user-specification of a content entity to move and target location for the specified content entity.

10. The method of claim 9, further comprising the step of providing a user interface communicating with the data repository, and providing a mechanism for enabling a user to select a content entity to move and specify the target location in the content object through the user interface.

11. A method for reordering content in a hierarchically structured content object stored as a plurality of hierarchically related content entities in a data repository, each content entity having an identifier, comprising the steps of:

defining the content object with an outline of containers and content entity identifiers such that moving a container or content entity identifier to a new location within the outline redefines the order of the object's content entities,

wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities.

12. The method of claim 11, further comprising the step of receiving a user-specification of a content entity to move and target location for the specified content entity.

13. The method of claim 12, further comprising the step of providing a user interface communicating with the data repository, and providing a mechanism for enabling a user to select a content entity to move and specify the target location in the content object through the user interface.

14. The method of claim 11, wherein the content object comprises a book, the content entities comprise sections and the containers comprise chapters and books.

15. A system for reordering content in a content object stored as a plurality of hierarchically related content entities in a data repository, each content entity having an identifier, comprising:

means for defining the content object with a list of content entity identifiers such that moving a content entity identifier to a new location within the list redefines the order of the object's content entities,

wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities.

16. The system of claim 15, further comprising means for receiving a user-specification of a content entity to move and target location for the specified content entity.

17. The system of claim 16, further comprising a user interface communicating with the data repository, and a mechanism for enabling a user to select a content entity to move and specify the target location in the content object through the user interface.

18. A system for reordering content in a hierarchically structured content object stored as a plurality of hierarchically related content entities in a data repository, each content entity having an identifier, comprising:

means for defining the content object with an outline of containers and content entity identifiers such that moving a container or content entity identifier to a new location within the outline redefines the order of the object's content entities,

wherein the hierarchically related content entities further comprise a parent container type and a child container type, wherein parent containers can contain child containers, and child containers can contain content entities.

19. The system of claim 18, further comprising means for receiving a user-specification of a content entity to move and target location for the specified content entity.

20. The system of claim 19, further comprising a user interface communicating with the data repository, and a mechanism for enabling a user to select a content entity to move and specify the target location in the content object through the user interface.

21. The system of claim 18, wherein the content object comprises a book, the content entities comprise sections and the containers comprise chapters and books.

22. The method of claim 1, further comprising calculating a cost for the content object according to costs of the content entities.

23. The method of claim 4, further comprising calculating a cost for the content object according to costs of the content entities.

24. The method of claim 8, further comprising calculating a cost for the content object according to costs of the content entities.

25. The method of claim 11, further comprising calculating a cost for the content object according to costs of the content entities.

26. The system of claim 15, further comprising means for calculating a cost for the content object according to costs of the content entities.

27. The system of claim 18, further comprising means for calculating a cost for the content object according to costs of the content entities.

28. The method of claim 1, wherein each content entity has an identifier with a format of parentcontainerref.childcontainerref.contententityref, where parentcontainerref is a reference to a parent container, childcontainerref is a reference to a child container and contententityref is a reference to a content entity, thereby indicating the hierarchical level of the entity.

29. The method of claim 28, wherein the parent container type is a book, the child container type is a chapter, and the content entity is a section.

30. The method of claim 4, wherein each content entity has an identifier with a format of parentcontainerref.childcontainerref.contententityref, where parentcontainerref is a reference to a parent container, childcontainerref is a reference to a child container and contententityref is a reference to a content entity, thereby indicating the hierarchical level of the entity.

31. The method of claim 30, wherein the parent container type is a book, the child container type is a chapter, and the content entity is a section.

32. The method of claim 8, wherein each content entity has an identifier with a format of parentcontainerref.childcontainerref.contententityref, where parentcontainerref is a reference to a parent container, childcontainerref is a reference to a child container and contententityref is a reference to a content entity, thereby indicating the hierarchical level of the entity.

33. The method of claim 32, wherein the parent container type is a book, the child container type is a chapter, and the content entity is a section.

34. The method of claim 11, wherein each content entity has an identifier with a format of parentcontainerref.childcontainerref.contententityref, where parentcontainerref is a reference to a parent container, childcontainerref is a reference to a child container and contententityref is a reference to a content entity, thereby indicating the hierarchical level of the entity.

35. The method of claim 34, wherein the parent container type is a book, the child container type is a chapter, and the content entity is a section.

36. The system to claim 15, wherein each content entity has an identifier with a format of parentcontainerref.childcontainerref.contententityref, where parentcontainerref is a reference to a parent container, childcontainerref is a reference to a child container and contententityref is a reference to a content entity, thereby indicating the hierarchical level of the entity.

37. The system of claim 36, wherein the parent container type is a book, the child container type is a chapter, and the content entity is a section.

38. The system of claim 18, wherein each content entity has an identifier with a format of parentcontainerref.childcontainerref.contententityref, where parentcontainerref is a reference to a parent container, childcontainerref is a reference to a child container and contententityref is a reference to a content entity, thereby indicating the hierarchical level of the entity.

39. The system of claim 38, wherein the parent container type is a book, the child container type is a chapter, and the content entity is a section.

40. A method for reordering content in a content object stored as a plurality of hierarchically related content entities in a data repository, each content entity having an identifier, comprising:

defining the content object with a first list of content entity identifiers such that moving a content entity identifier from the first list of content entity identifiers to a new location within the first list of content entity identifiers redefines the order of the object's content entities,

wherein moving a content entity identifier in the first list to a new location comprises:

selecting the content entity identifier from the first list of content entity identifiers to be moved; and

specifying a location from a second list of content entity identifiers where the content entity identifier from the first list of content entity identifiers is to be moved;

wherein the specified location comprises at least one of a current content entity identifier or a newly created content entity identifier.

41. A method for reordering content in a content object according to claim 40, wherein the specified location from the second list of content entity identifiers is selected via a drop down box.

42. A method for reordering content in a customized electronic book stored as a plurality of hierarchically related content entities in a computer database, each content entity having an identifier, comprising:

defining the customized book with a first outline of content entity identifiers such that moving a content entity identifier from the first outline to a new location within the first outline of content entity identifiers redefines the order of the content entities within the customized book,

wherein moving a content entity identifier from the first outline to a new location comprises:

selecting a content entity identifier from the first outline to be moved; and

specifying a location from a second outline of content entity identifiers where the content entity identifier of the first outline is to be moved;

wherein the specified location comprises at least one of a current content entity identifier or a newly created content entity identifier.

43. A method for reordering content in a customized electronic book according to claim 42, wherein the specified location from the second outline of content entity identifiers is selected via a drop down box.

**APPEAL BRIEF UNDER 37 C.F.R. § 41.37**  
**U.S. Application No. 09/488,971**

**Docket No. A8519**

**EVIDENCE APPENDIX:**

None.

**APPEAL BRIEF UNDER 37 C.F.R. § 41.37**  
**U.S. Application No. 09/488,971**

**Docket No. A8519**

**RELATED PROCEEDINGS APPENDIX**

None.